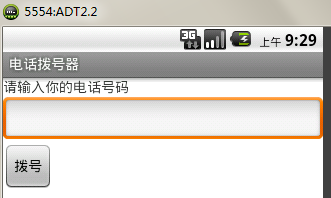
Android电话拨号和发送短信

一：电话拨号：



需要设计出如图的拨号器

1. 设计主界面 Layout –》activity\_main.xml

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:orientation=*"vertical"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"* >

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/hello\_world"* />

<EditText

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:id=*"@+id/editText"* />

<Button

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/call"*

android:id=*"@+id/button"*/>

</LinearLayout>

LinearLayout (线性布局)、AbsoluteLayout(绝对布局)、RelativeLayout(相对布局)、TableLayout(表格布局)、FrameLayout(帧布局)

默认是RelativeLayout布局，目前还不知道怎么布局，知道线性布局有水平线性和竖直线性

Android：id是为了设置能拿到EditText和Button的数据而设置的唯一标示，在R中可以有定义，从而拿到按钮的触发条件和编辑框的文本

1. String.xml

<string name=*"app\_name"*>电话拨号器</string>

<string name=*"action\_settings"*>Settings</string>

<string name=*"hello\_world"*>请输入你的电话号码</string>

<string name=*"call"*>拨号</string>

1. MainActivity.java

**private** EditText editText;

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

Button button = (Button) findViewById(R.id.*button*);

editText = (EditText) findViewById(R.id.*editText*);

button.setOnClickListener(**new** View.OnClickListener(){

**public** **void** onClick(View v) {

String number = editText.getText().toString();

**Intent intent = null;//使用Intent目的是为了激活Activity**

**if**(!"".equals(number) && number != **null**){//还需要用正则表达要求只能是数字，当然实际上拨号一定是数字盘

intent = **new** Intent(Intent.*ACTION\_CALL*,Uri.*parse*("tel:"+number));

startActivity(intent);//开启服务

}

// intent.setAction(Intent.ACTION\_CALL);

// intent.setData(Uri.parse("tel:"+number));

//startActivity(intent);

// 方法的内部会自动为intent对象设置类别:android.intent.category.DEFAULT

}

});

}

**由于我们可以通过findViewById(R.id.---)从R中拿到Button和EditText标示，而用button.setOnClickListener(内部类 必须要实现OnClickListener)来实现点击button而触发监听，从而实现button的功能，而EditText只需要输出数据，在通过Intent来激活Activity控件，用在Intent.ACTION\_CALL激活了android中拨号控件，Uri只是输入数字**

**intent = new Intent(Intent.*ACTION\_CALL*,Uri.*parse*("tel:"+number))，startActivity(intent);开启服务**

1. AndroidManifest.xml

<uses-permission

android:name=*"android.permission.CALL\_PHONE"*/>

**由于android安装是为了保护用户的隐私，而有涉及权限，当我们安装在手机上提示涉及来用户隐私，可以不安装，由于模拟器中默认安装运行就没有提示**

**下面是模拟过程**

**二：发送短信**



**1.** 设计主界面 Layout –》activity\_main.xml

<LinearLayout xmlns:android=*"http://schemas.android.com/apk/res/android"*

android:orientation=*"vertical"*

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"fill\_parent"* >

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/phone"* />

<EditText

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:id=*"@+id/editText1"* />

<TextView

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/content"* />

<EditText

android:layout\_width=*"fill\_parent"*

android:layout\_height=*"wrap\_content"*

android:minLines=*"3"*

android:id=*"@+id/editText2"* />

<Button

android:layout\_width=*"wrap\_content"*

android:layout\_height=*"wrap\_content"*

android:text=*"@string/sendSMS"*

android:id=*"@+id/button"*/>

</LinearLayout>

2．文本values -- > String.xml

<string name=*"app\_name"*>短信发送器</string>

<string name=*"action\_settings"*>Settings</string>

<string name=*"phone"*>请输入你的电话号码</string>

<string name=*"content"*>请输入短信内容</string>

<string name=*"sendSMS"*>发送短信</string>

**3. 主要代码**

**private** EditText editText1;

**private** EditText editText2;

**protected** **void** onCreate(Bundle savedInstanceState) {

**super**.onCreate(savedInstanceState);

setContentView(R.layout.*activity\_main*);

editText1 = (EditText)**this**.findViewById(R.id.*editText1*);

editText2 = (EditText)**this**.findViewById(R.id.*editText2*);

Button button = (Button)**this**.findViewById(R.id.*button*);

button.setOnClickListener(**new** SendMsgOnClickListener());

}

**private** **final** **class** SendMsgOnClickListener **implements** View.OnClickListener{

**public** **void** onClick(View v) {

String phone = editText1.getText().toString();

String content = editText2.getText().toString();

**SmsManager smser = SmsManager.*getDefault*();**

**ArrayList<String> contents = smser.divideMessage(content);**

**for(String msg : contents){**

**smser.sendTextMessage(phone, null, msg, null, null);**

**}**

**Toast.*makeText*(MainActivity.this, "发送成功", 1).show();** }

权限：[file:///D:/android-windows/sdk/docs/reference/android/Manifest.permission.html](file:///D:\\android-windows\\sdk\\docs\\reference\\android\\Manifest.permission.html)

<uses-permission android:name=*"android.permission.SEND\_SMS"*/>

**解释：**

此项与拨号是差不多的，所以只需要解释不同之处，其中，SmsManager是android提供SmsManager接口，实现短信的驱动的接口，而**smser.divideMessage(content);是由于短信文字数量要求一定范围内，所以android通过divideMessage（内容）来截取字符，当超过140字符以内就在同一地址，用集合ArrayList来存储**

**用增强循环方式，来发送，smser.sendTextMessage(phone, null, msg, 发送后得到发送状态, 得到是否收到状态);**

**Toast.*makeText*(MainActivity.this, "发送成功", 1).show();用Toast方式来接收是否发送成功后的提示**

**提示有三种：状态栏 应用与应用服务、广播等 不需要软件界面**

**对话框 重要信息的使用**

**Toast 一般信息**

